Bipartite Modular Multiplication

Marcelo E. Kaihara
and
Naofumi Takagi

Department of Information Engineering
Nagoya University

Outline

- Background and Objective
- Preliminaries
  - Ordinary Modular Multiplication
  - Montgomery Multiplication
- New Method
- Hardware Implementation
- Summary

Background and Objective

- Modular Multiplication
  - Basic operation in public-key cryptographic applications.
- Fast method required
  - Operation with large integers (huge amount of computation)
  - A fast method enables: The use of large keys and real time decryption.

Develop fast method for calculating modular multiplication

Main Idea

Multiplier is split into two parts

Ordinary Multiplication

Interleaved Modular Multiplication Algorithm (classical method)

Process in parallel to boost speed

Montgomery Multiplication

Montgomery Multiplication Algorithm proposed by P.L.Montgomery, 1985

Ordinary Modular Multiplication

Definition:

$$M: \text{modulus} \qquad X, Y \in Z_M = \{0, 1, \cdots, M-1\}$$

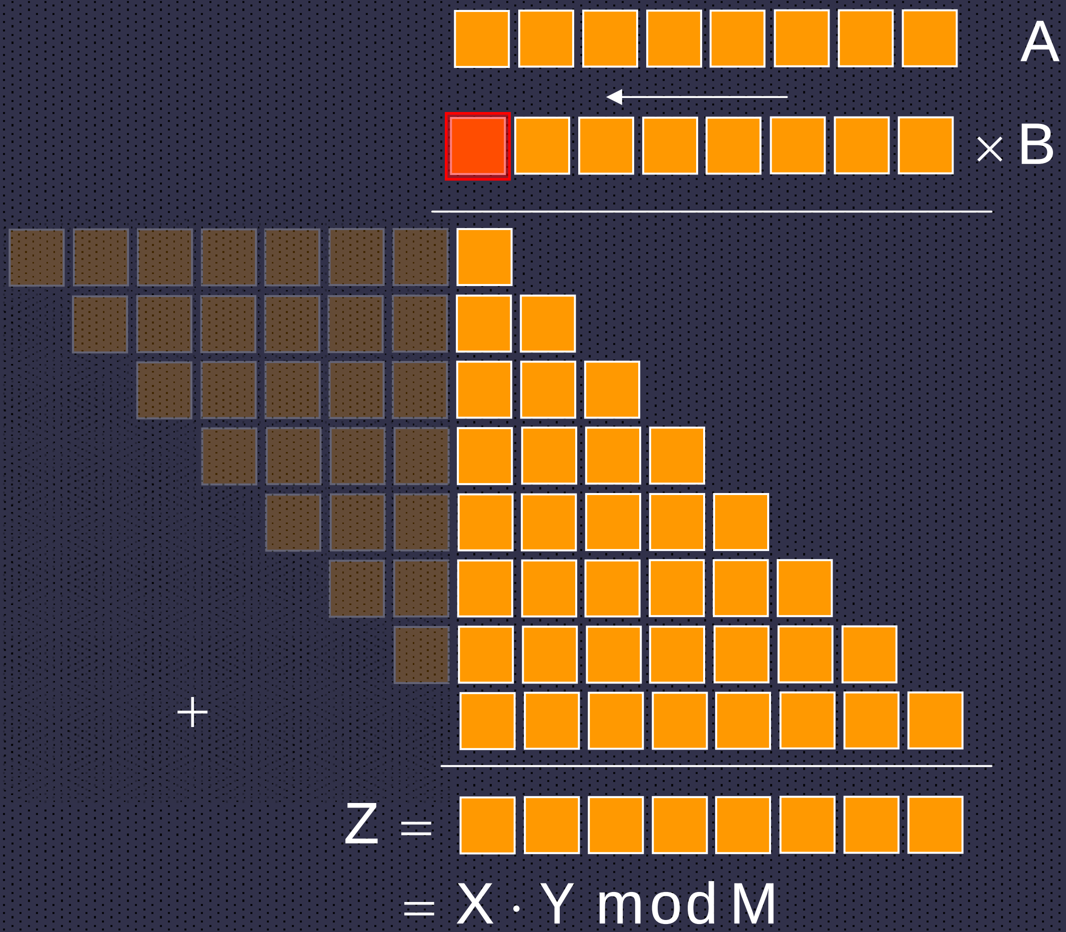
$$X \times Y \triangleq X \cdot Y \mod M$$

Multiprecision arithmetic:

$$r = 2^k, \quad M = \sum_{i=0}^{n-1} m_i \cdot r^i, \quad X = \sum_{i=0}^{n-1} x_i \cdot r^i, \quad Y = \sum_{i=0}^{n-1} y_i \cdot r^i$$

Ordinary Modular Multiplication

Interleaved Modular Multiplication
Process of Computation



Algorithm

$A := X; B := Y; M := M;$

$S := 0;$

for $i := n - 1$ downto 0 do

$\quad S := r \cdot S + b_{n-1} A;$

$\quad q_C := \lfloor S / M \rfloor;$

$\quad S := S - q_C \cdot M;$

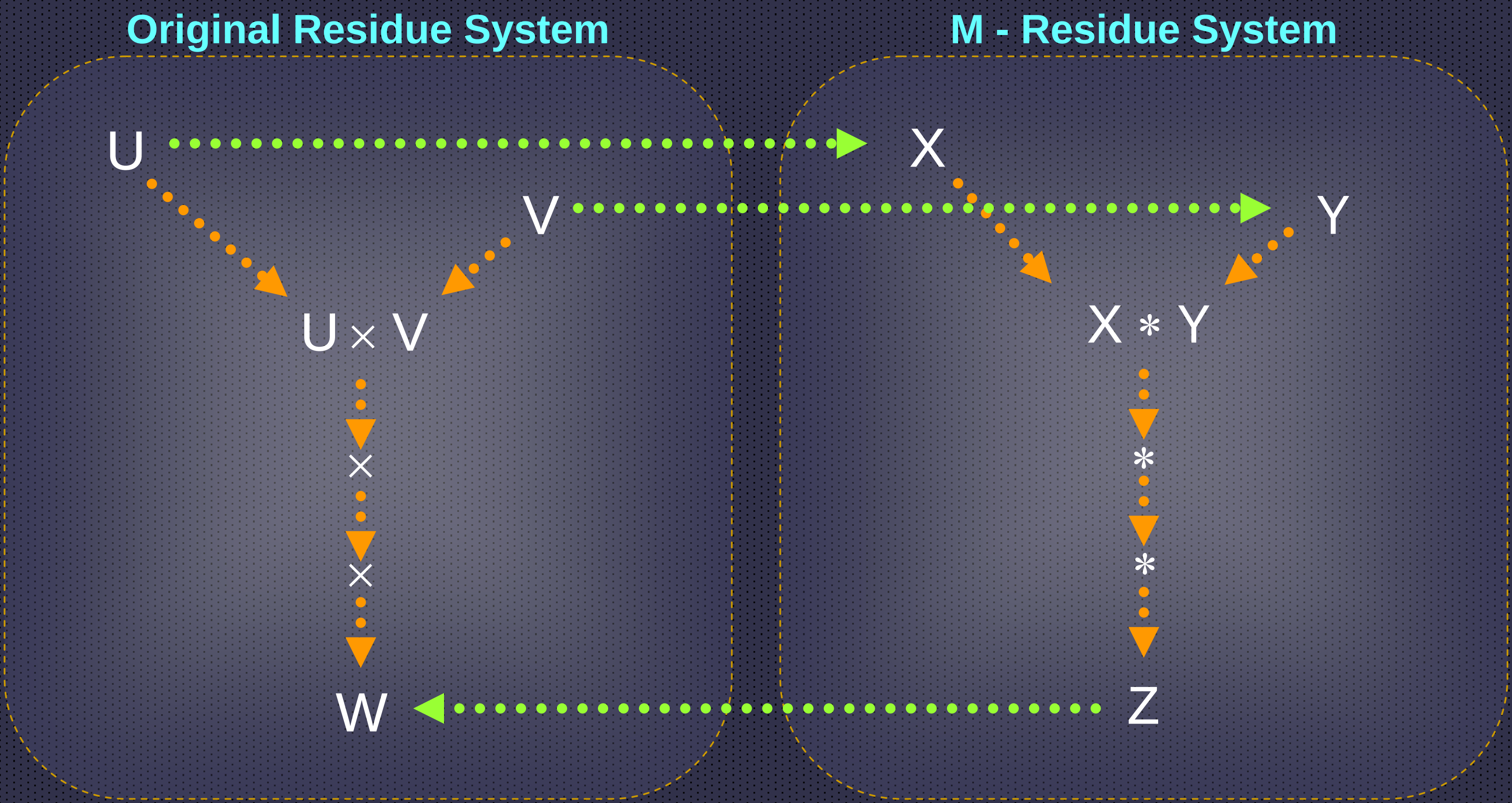
$\quad B := r \cdot B;$

endfor

$Z := S;$

Montgomery Multiplication

M-Residue System



Chained multiplications (in modular exponentiation) are performed in the M-Residue system

Montgomery Multiplication

Definition:

$M : n\text{-word}$, $\gcd(r, M) = 1$, $R_M = r^n > M$

$X, Y \in Z_M = \{0, 1, \cdots, M-1\}$

$$X * Y \triangleq X \cdot Y \cdot r^{-n} \mod M$$

Montgomery Multiplication

Digit-serial Montgomery Algorithm
Process of Computation

Algorithm

$A := X; B := Y; M := M;$

$T := 0;$

for $i := 0$ to $n - 1$ do

  $T := T + b_0 \cdot A;$

  $q_M := (-t_0 \cdot m_0^{-1}) \bmod r;$

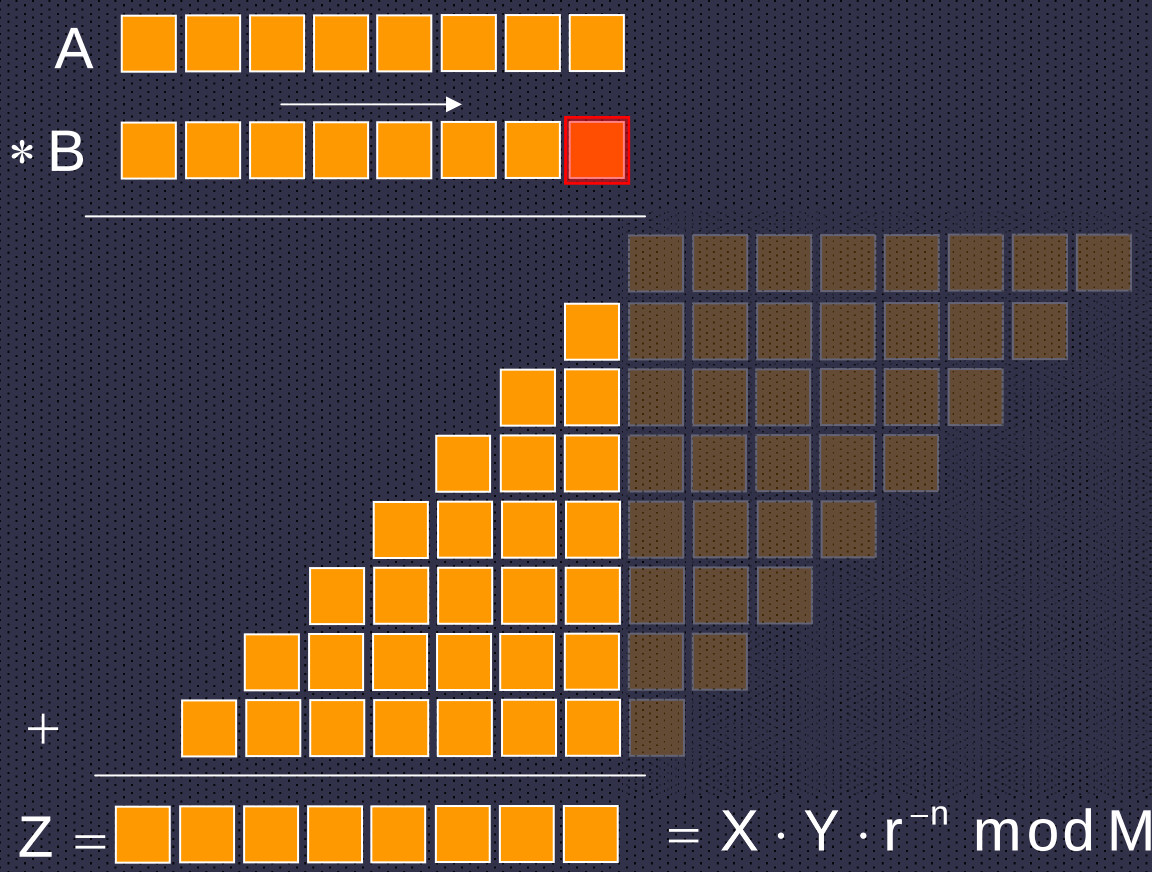
  $T := (T + q_M \cdot M) / r;$

  $B := B / r;$

endfor

if $T \geq M$ then $Z := T - M;$

else $Z := T;$



New Modular Multiplication

Operands are transformed into a new residue system

Multiplier is split into two parts

Ordinary Multiplication		Montgomery Multiplication
Interleaved Modular Multiplication Algorithm (classical method)	Process in parallel to boost speed	Montgomery Multiplication Algorithm proposed by P.L.Montgomery, 1985

Result in the same residue system

New Modular Multiplication

A lot of research to speed up both algorithms

Ordinary Multiplication

Interleaved Modular Multiplication Algorithm
(classical method)

Montgomery Multiplication

Montgomery Multiplication Algorithm
proposed by P.L.Montgomery, 1985

Take advantage of developed techniques

Halve the number of iteration

Double the speed

New Modular Multiplication

New transformation constant $R = r^{\alpha n} < M$

$\alpha : \alpha \in \mathbb{Q},\ 0 < \alpha < 1,\ \alpha \cdot n \in Z$

Original Residue System	New Residue System
$(Z_M, \times)$	$(Z'_M, \circledast)$
U	$X = U \cdot r^{\alpha n} \bmod M$
$U \times V = U \cdot V \bmod M$	$X \circledast Y = X \cdot Y \cdot r^{-\alpha n} \bmod M$
V	$Y = V \cdot r^{\alpha n} \bmod M$

Isomorphic

New Modular Multiplication

Definition:

$$M : n\text{-word}, \quad \gcd(r, M) = 1, \quad R = r^{\alpha n} < M$$

$$\alpha : \alpha \in \mathbb{Q}, \quad 0 < \alpha < 1, \alpha \cdot n \in \mathbb{Z}$$

$$X, Y \in Z_M = \{0, 1, \cdots, M-1\}$$

$$X \circledast Y \triangleq X \cdot Y \cdot r^{-\alpha n} \mod M$$

Computation of the New Modular Multiplication

$$X \circledast Y \triangleq X \cdot Y \cdot r^{-\alpha n} \bmod M$$

$$Y_H \cdot r^{\alpha n} + Y_L$$

$$= X \cdot (Y_H \cdot r^{\alpha n} + Y_L) \cdot r^{-\alpha n} \bmod M$$

$$= X \cdot Y_H \cdot \cancel{r^{\alpha n}} \cdot \cancel{r^{-\alpha n}} + X \cdot Y_L \cdot r^{-\alpha n} \bmod M$$

$$= X \cdot Y_H + X \cdot Y_L \cdot r^{-\alpha n} \bmod M$$

Computation of the New Modular Multiplication

$$X \circledast Y = \underbrace{X \cdot Y_H}_{} + \underbrace{X \cdot Y_L \cdot r^{-\alpha n}}_{} \mod M$$

Interleaved Modular Multiplication Algorithm

Montgomery Multiplication Algorithm

New Modular Multiplication

[Algorithm KT]

Input: $M : r^{n-1} < M < r^{n}$, M odd

$X, Y \in Z'_M$

Output: $Z = X \cdot Y \cdot r^{-\alpha n} \bmod M \quad (Z \in Z'_M)$

Algorithm:

Step 1: $A := X;\ M := M;\ S := 0;\ T := 0;$

$B_H := Y_H;\ B_L := Y_L$

Step 2: $\{\ S := \mathrm{Interleaved\_modmul}(A, B_H);$

$T := \mathrm{Montgomery\_modmul}(A, B_L);\ \}$

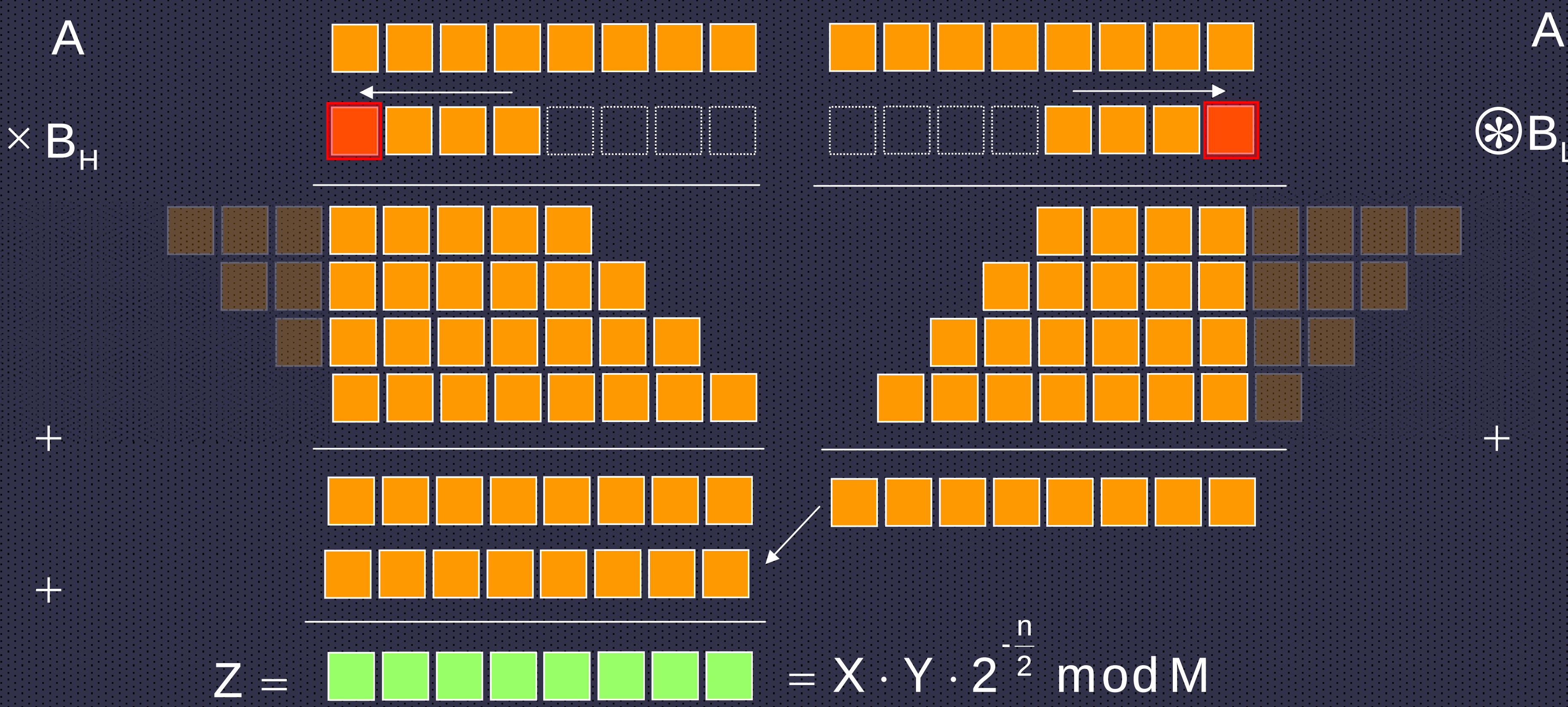
Step 3: $Z := (S + T) \bmod M;$

New Modular Multiplication

Process of Computation (α=1/2)

The multiplier is processed from both sides in parallel

$$X \circledast Y = X \cdot Y_H + X \cdot Y_L \cdot r^{-n/2} \bmod M$$



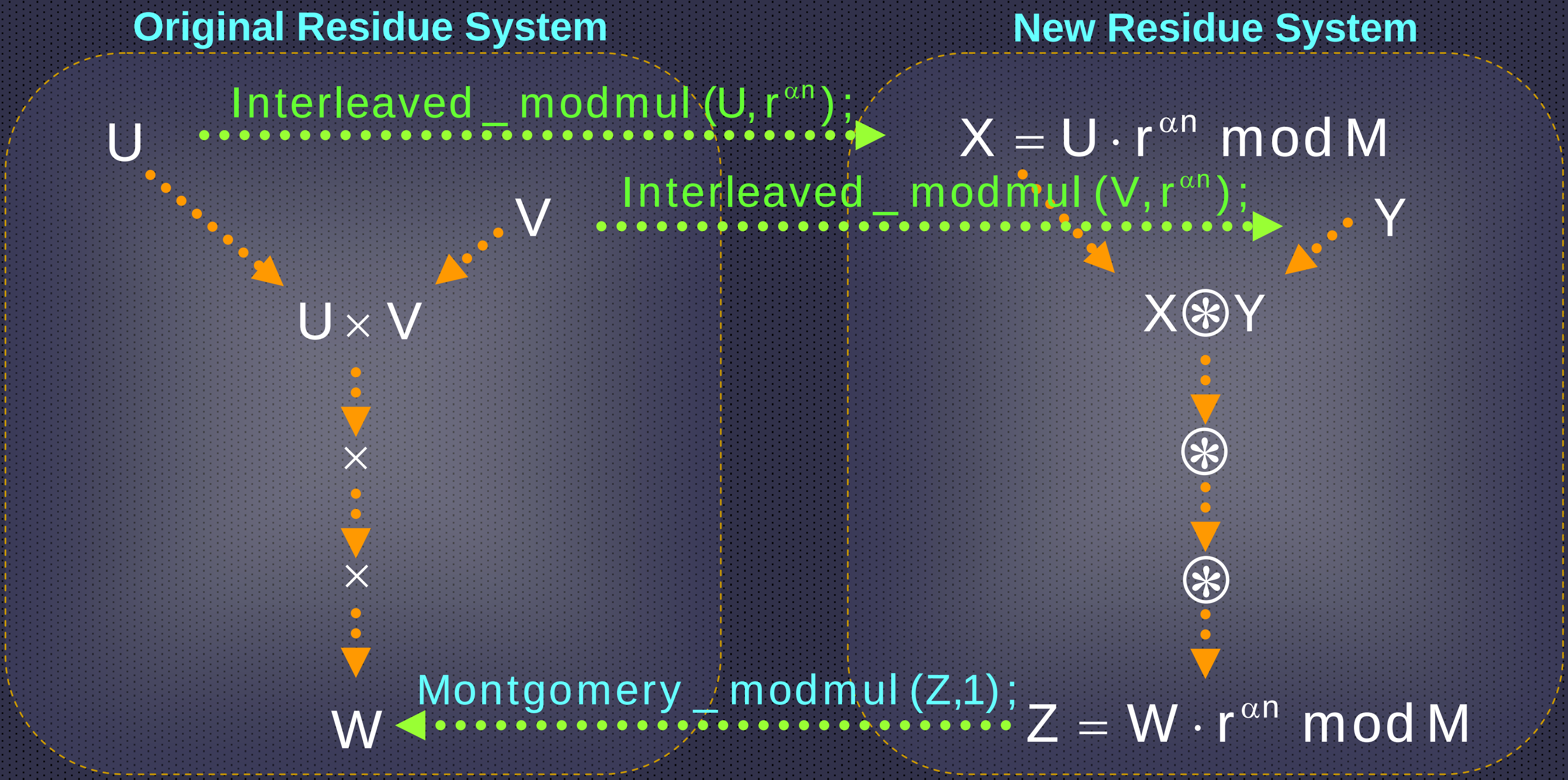
$$= X \cdot Y \cdot 2^{-\frac{n}{2}} \bmod M$$

New Modular Multiplication

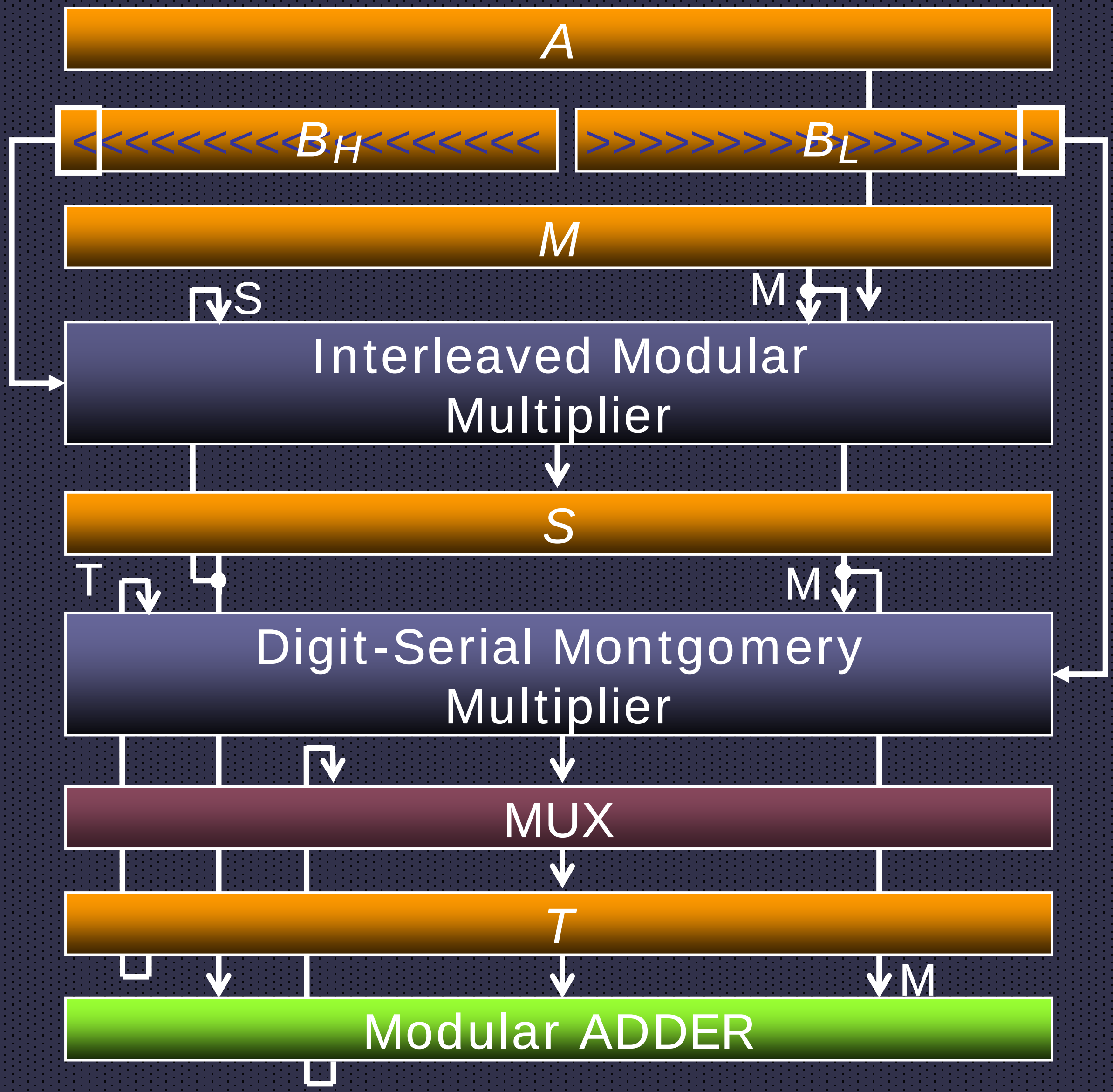
Conversions between residue systems

Conversions can be done in half the time

No need for pre-computed constants



Hardware Implementation

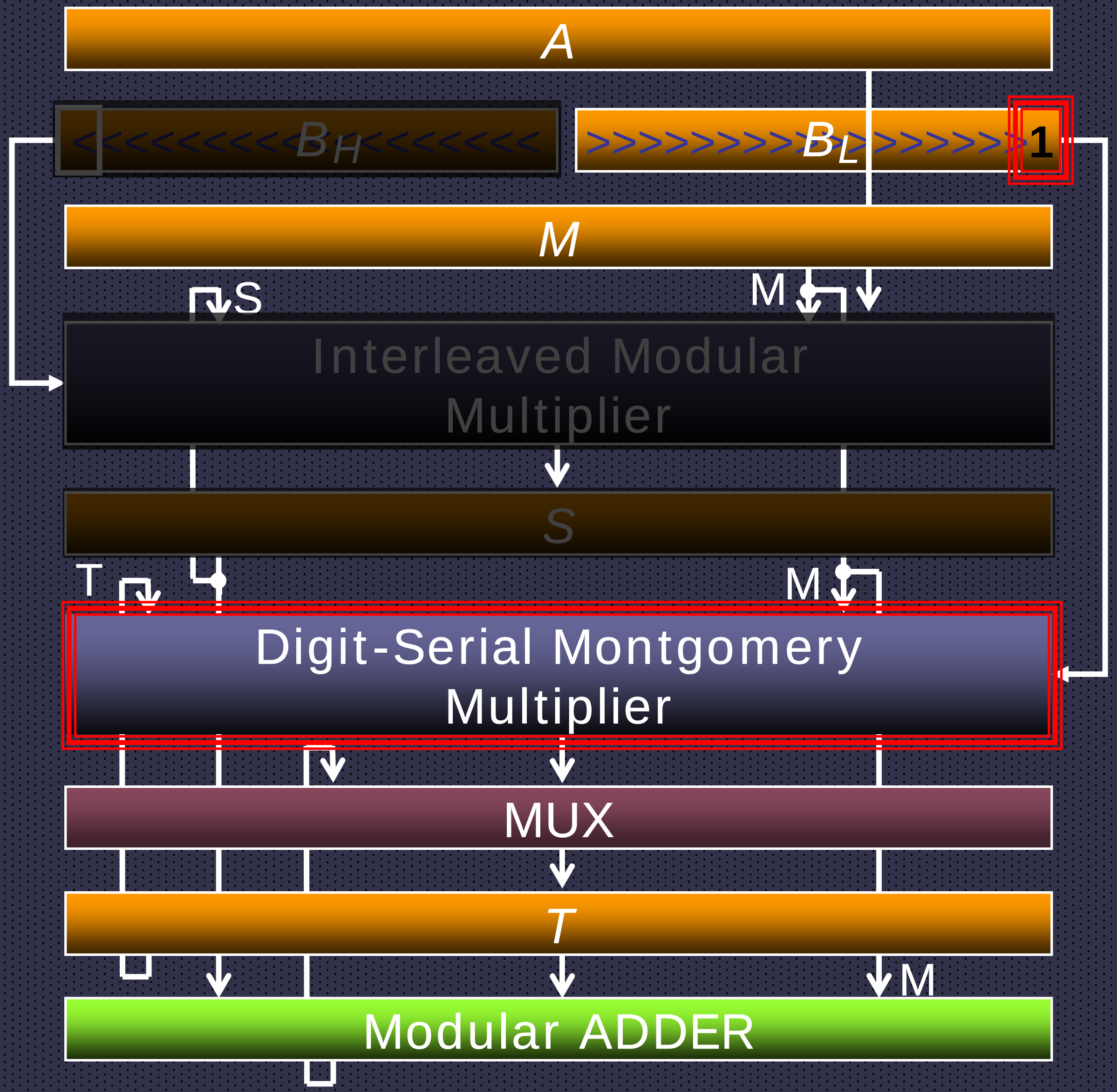


Hardware Implementation

From the Original to the New Residue System

Hardware Implementation

From the New to the Original Residue System



Hardware Implementation

Characteristics of the Circuit Based on the New Algorithm

- Can be constructed using already designed circuits of lower radix.
- Amount of hardware proportional to n.
- When using multipliers of similar performance (α=1/2), execution time n/2+1 clk cycles, i.e. acceleration twice the speed of the original multipliers.

Hardware Implementation

Different Combination of Multipliers

By changing α it is possible to use different combinations of multipliers

Interleaved Modular Multiplication

Radix-2

Multipliers of different performance

Montgomery Multiplier

Radix-4

Summary

- We proposed a new computation method for speeding up modular multiplication. Multiplier processed from both sides in parallel.
- With multipliers of similar performance, number of clock cycles halved. Multipliers of different performance can be used by changing the value of α .
- The proposed method suitable for both hardware implementation; and software implementation in a multiprocessor environment.
- The technique used in the proposed method can be adapted for operation in the binary extended field $GF(2^m)$.